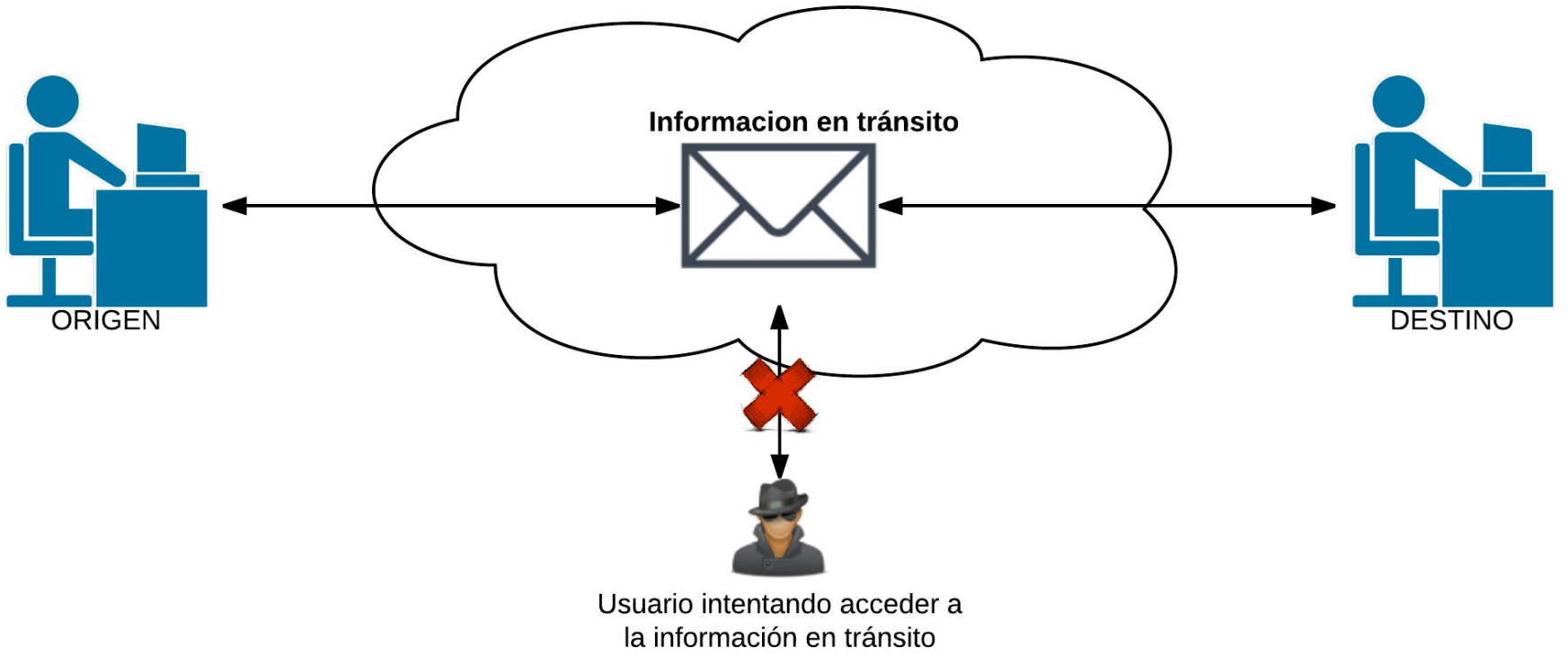
Seguridad de Redes

- **Requerimientos**
- **Integridad**
- **Autenticidad**
- **Sistemas de cifrado**

Objetivo

- Seguridad de redes puede significar muchas cosas, entre ellas:
  - Control de Acceso a la red
  - “Hardening” de equipos en la red
  - Filtrado de trafico -> “firewalls”
  - Deteccion / Prevencion de intrusos
  - **Asegurar la información mientras atraviesa la red**

Objetivo (cont.)



Requerimientos

Requerimientos (cont.)

Requerimientos (cont.)

CONFIDENCIALIDAD

La información transmitida, aunque sea capturada, no puede ser comprendida

Requerimientos (cont.)

INTEGRIDAD

La información transmitida no se modificó en el camino

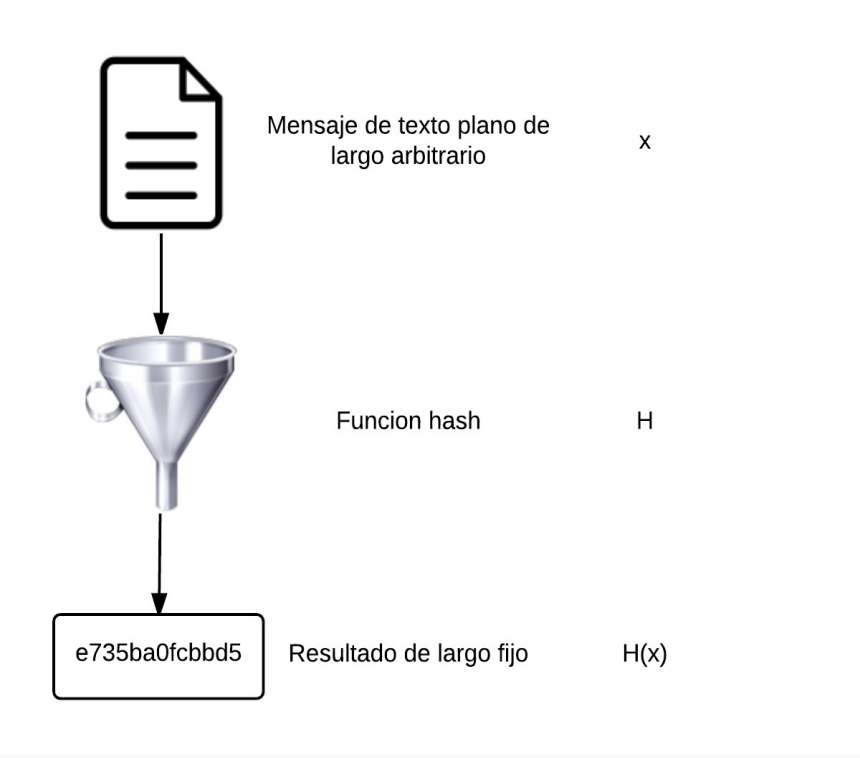
Requerimientos (cont.)

AUTENTICACION

La información transmitida proviene de quien dice ser

Integridad

- Para asegurar la integridad se utilizan funciones hash
- Funcion matematica en un solo sentido. Similar al CRC



Integridad (cont.)

Pagar a Tavo		$100.00
Cien pesos		
39a32eb8fc		
Hash original		

Pagar a Edu		$1000.00
Mil Pesos		
0fa8264cd1		
Hash en destino		

Se intercepta el mensaje y se modifica

Se debe comparar el hash computado en el destino con el computado en el origen!

Integridad (cont.)

- Funciones de hash
  - MD5 → valor hash de 128 bits
  - SHA-1 → valor hash de 160 bits
  - SHA-256 → valor hash de 256 bits

- http://en.wikipedia.org/wiki/Comparison_of_cryptographic_hash_functions

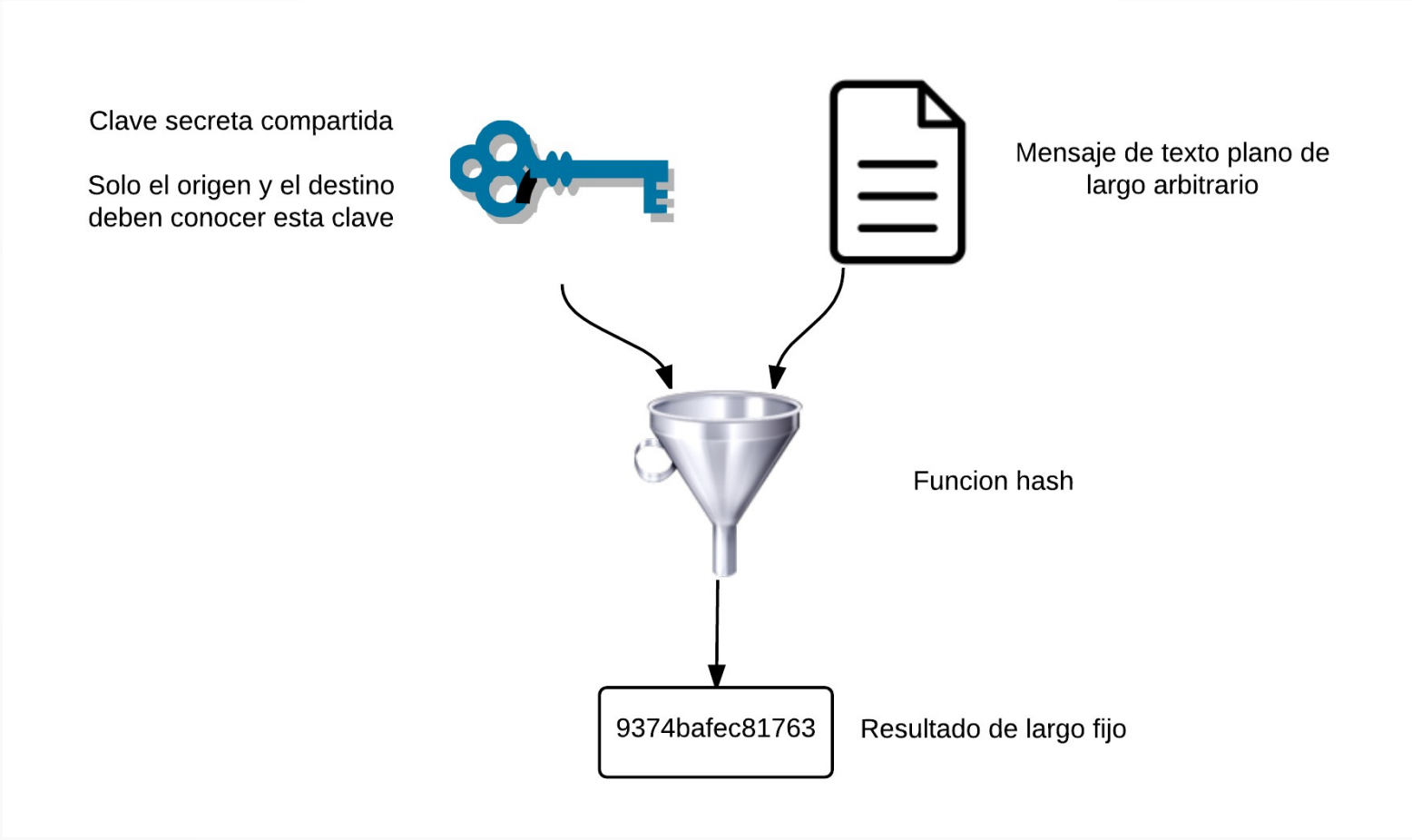
Integridad (cont.)

- Laboratorio hashes
  - Crear un archivo de texto
  - Obtener el hash de ese archivo
  - Modificar el archivo de texto
  - Obtener el hash del texto modificado y comparar con el valor anterior
  - Comparar con diferentes metodos de hashing
  - Comandos:
    - `md5sum`
    - `sha1sum`
    - `sha256sum`

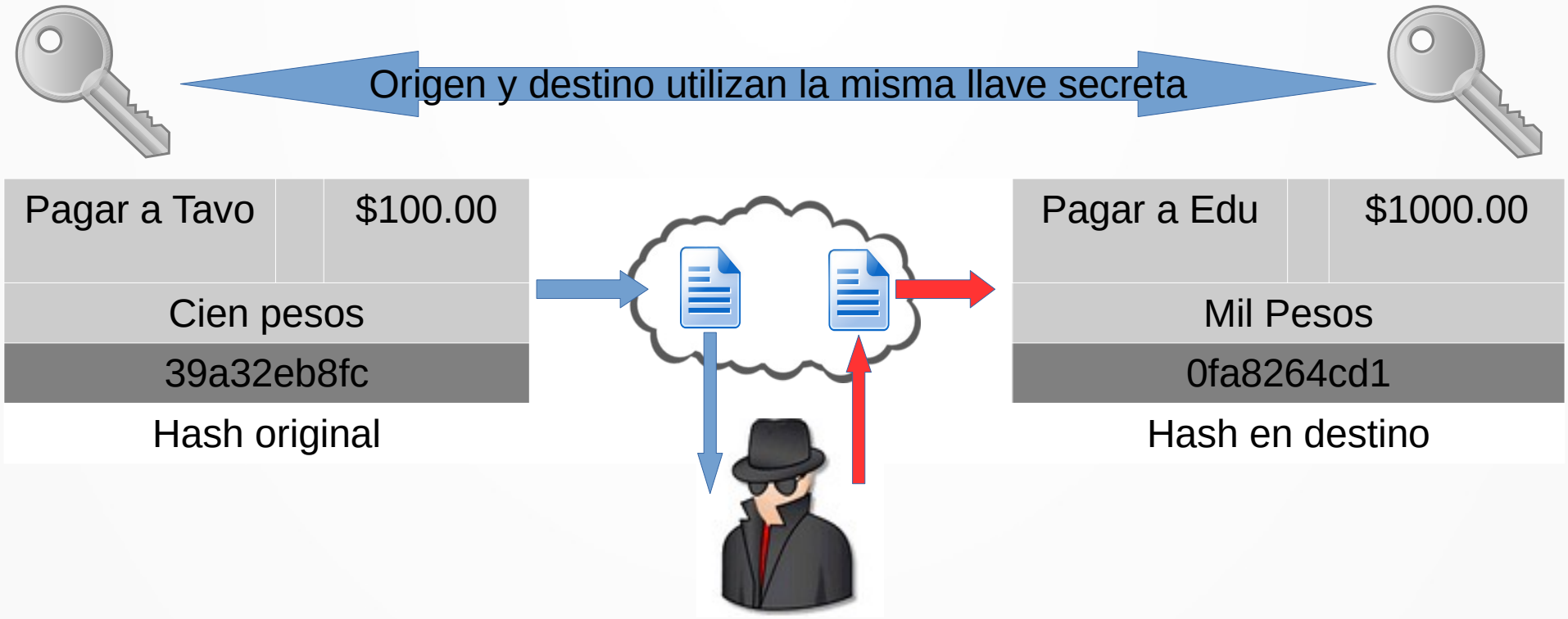
Integridad + Autenticacion

- Como asegurar que el mensaje no fue alterado (integridad)
- Y proviene de un origen conocido (autenticidad)
- HMAC o KHMAC: Keyed-Hash Message Authentication Code
- Utilizar una llave (clave) compartida agregando autenticidad al hash
- La llave debe ser secreta. Solo debe ser conocida por el origen y el destino

Integridad + Autenticacion (cont.)



Integridad + Autenticacion (cont.)



Ya no es completamente necesario comparar el hash de destino con el de origen.
Solo se debe computar el hash en el destino utilizando la llave secreta.
El valor debe ser completamente diferente si se utilizó una llave falsa.

Integridad + Autenticacion (cont.)

- Laboratorio HMAC
  - Crear un archivo de texto
  - Crear una clave secreta
  - Generar un HMAC del archivo de texto utilizando la clave secreta
  - Verificar el HMAC si modifica el archivo de texto
  - Verificar el HMAC si modifica la clave secreta
  - Repoducir el procedimiento en diferentes tipos de hash (md5,sha1,sha256)
  - Comandos:
    - `openssl dgst -sha1 -hmac “clavesecreta” textoATransmitir.txt`

Gestión de llaves

- En un sistema criptográfico es una de las partes mas difíciles de lograr y mantener de manera segura
  - Generación de llave
  - Verificación de llave
  - Transferencia de llave
  - Almacenamiento de llave
  - Duración de la llave
  - Revocar / Eliminar la llave

Gestión de llaves

- Generación de llave
  - Puede ser elegida por el usuario
  - Por lo general es automatizada
  - Se necesita de un buen generador de números al azar
  - Largo/tamaño de llave, en bits.
    - Cuanto mas largo, mejor, pero también consume mayores recursos
  - Keyspace, el numero de posibilidades que se pueden generar con un largo especifico
  - A medida que crece el largo de la llave, el keyspace crece exponencialmente

https://www.keylength.com

Gestión de llaves

- Verificación de llave
  - Casi siempre existen llaves débiles
  - Se deben identificar y regenerar
  - Ej: Cifrado Cesar
    - la llave 0 o la 25 no cifran el mensaje

Gestión de llaves

- Transferencia de llave
  - Se necesita un mecanismo seguro de transferencia
  - Como se ponen de acuerdo ambos extremos de la comunicación
  - Seguramente haciendo uso de un medio inseguro

Gestión de llaves

- Almacenamiento de llave
  - Sistemas operativos modernos son multiusuario
  - La llave puede ser almacenada en memoria para rápido acceso / re-utilización
  - Que pasa si la pagina de memoria debe bajarse a disco?
  - Que mecanismos de protección se utilizan para la información en reposo?

Gestión de llaves

- Duración de la llave
  - Para mejorar la seguridad de los algoritmos de cifrado se debe utilizar una duración acotada
  - Algunos algoritmos utilizan una duración de 24 horas x defecto, ej IPsec
  - Se puede mejorar la seguridad si reducimos el tiempo, por ejemplo a 30 minutos

Gestión de llaves

- Revocar / Eliminar la llave
  - El proceso de revocación debe avisar a todos los interesados que la llave fue comprometida y no debe volver a usarse
  - El proceso de eliminación debe asegurarse que no exista manera de que se pueda recuperar una llave vieja

Sistemas de cifrado

Sistemas de transposición

Alterar el orden de los elementos del mensaje

Escítala

E	n		u	n		l	u	g	a
r		d	e		l	a		M	a
n	c	h	a	,		d	e		c
u	y	o		n	o	m	b	r	e
	n	o		q	u	i	e	r	o
	a	c	o	r	d	a	r	m	e

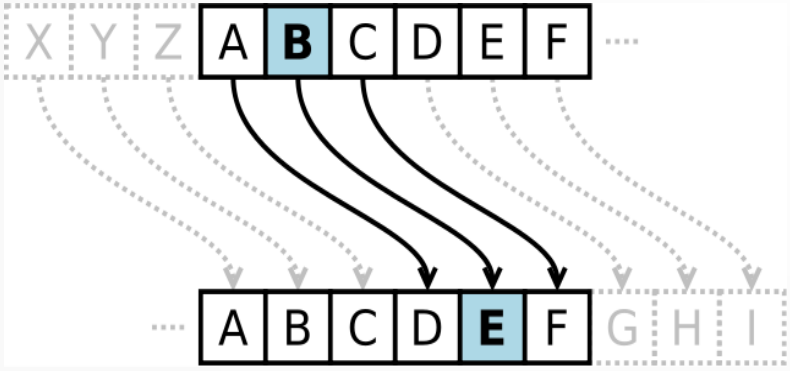
E	r	n	u		
n		c	y	n	a
	d	h	o	o	c
u	e	a			o
n		,	n	q	r
	l		o	u	d
l	a	d	m	i	a
u		e	b	e	r
g	M		r	r	m
a	a	c	e	o	e

Sistemas de cifrado (cont.)

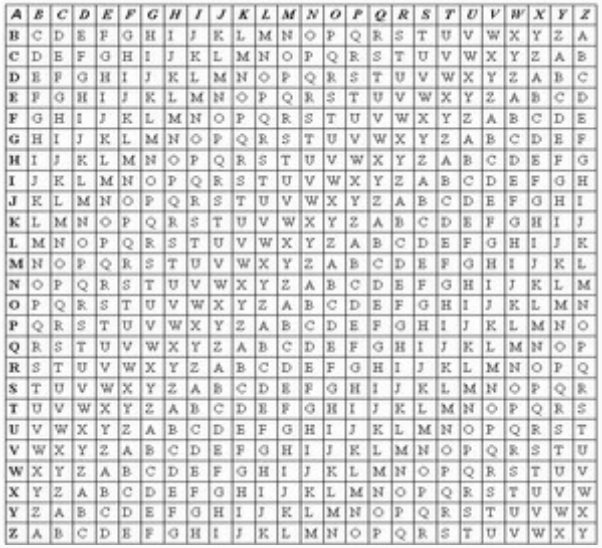
Sistemas de sustitución

Reemplazar un carácter por otro

Cifrado “cesar”
Por desplazamiento, n caracteres
Monoalfabético
Palabra clave: n desplazamientos



Cifrado “Vigenère”
Polialfabetico
Palabra clave se repite



```
mensaje:     P A R I S  V A U T  B I E N  U N E  M E S S E
clave:       L O U P L  O U P L  O U P L  O U P  L O U P L
criptograma:A O M X D  K U K E  P C T X  J H T  W S N I O
```

Sistemas de cifrado (cont.)

Sistemas de “relleno de un solo uso” (one-time pad)

Combinación de texto claro con clave aleatoria

Texto Plano	V	E	R	N	A	M	C	I	P	H	E	R
Equivalente numerico	21	4	17	13	0	12	2	8	15	7	4	17
+ Numero Aleatorio (CLAVE)	76	48	16	82	44	3	58	11	60	5	48	88
= sum	97	52	33	95	44	15	60	19	75	12	52	105
= mod 26	19	0	7	17	18	15	8	19	28	12	0	1
Texto Cifrado	T	A	H	R	S	P	I	T	X	M	A	B

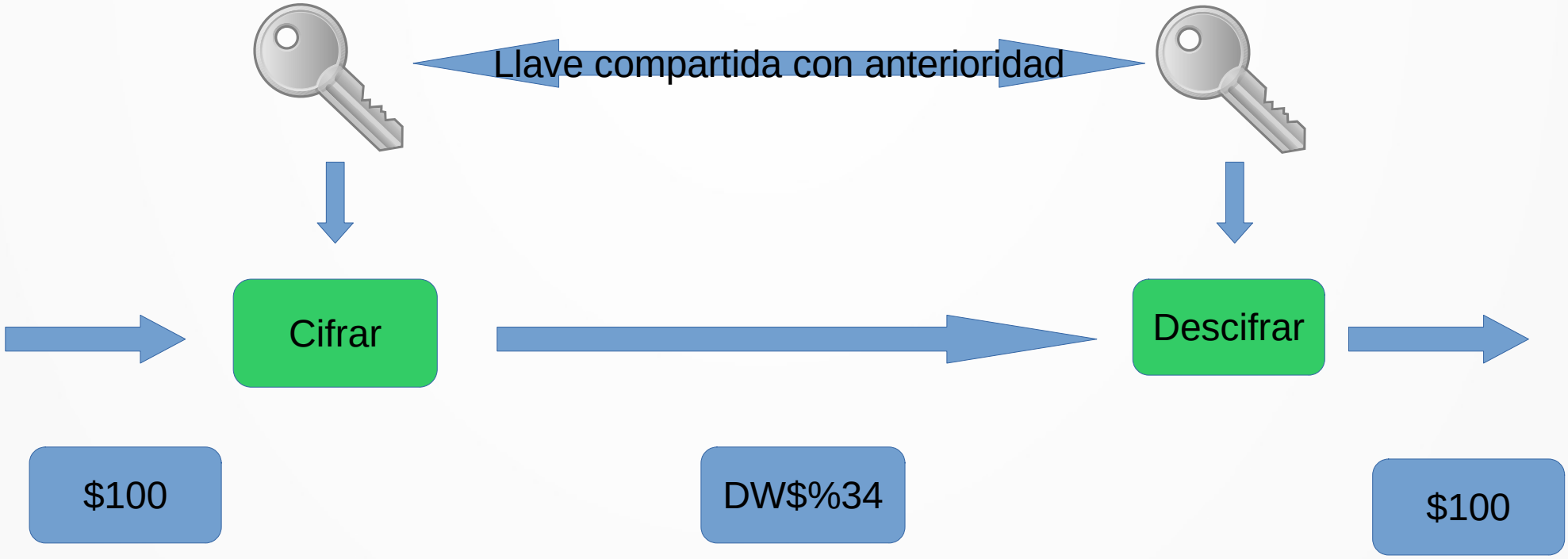
Sistemas de cifrado (cont.)

- Dos formas de proteger la seguridad de los datos cifrados
  - Proteger el algoritmo
    - Algoritmo secreto. Al ser revelado todos deben cambiar el algoritmo.
  - Proteger la llave
    - En la criptografía moderna todos los algoritmos son públicos

Sistemas de cifrado (cont.)

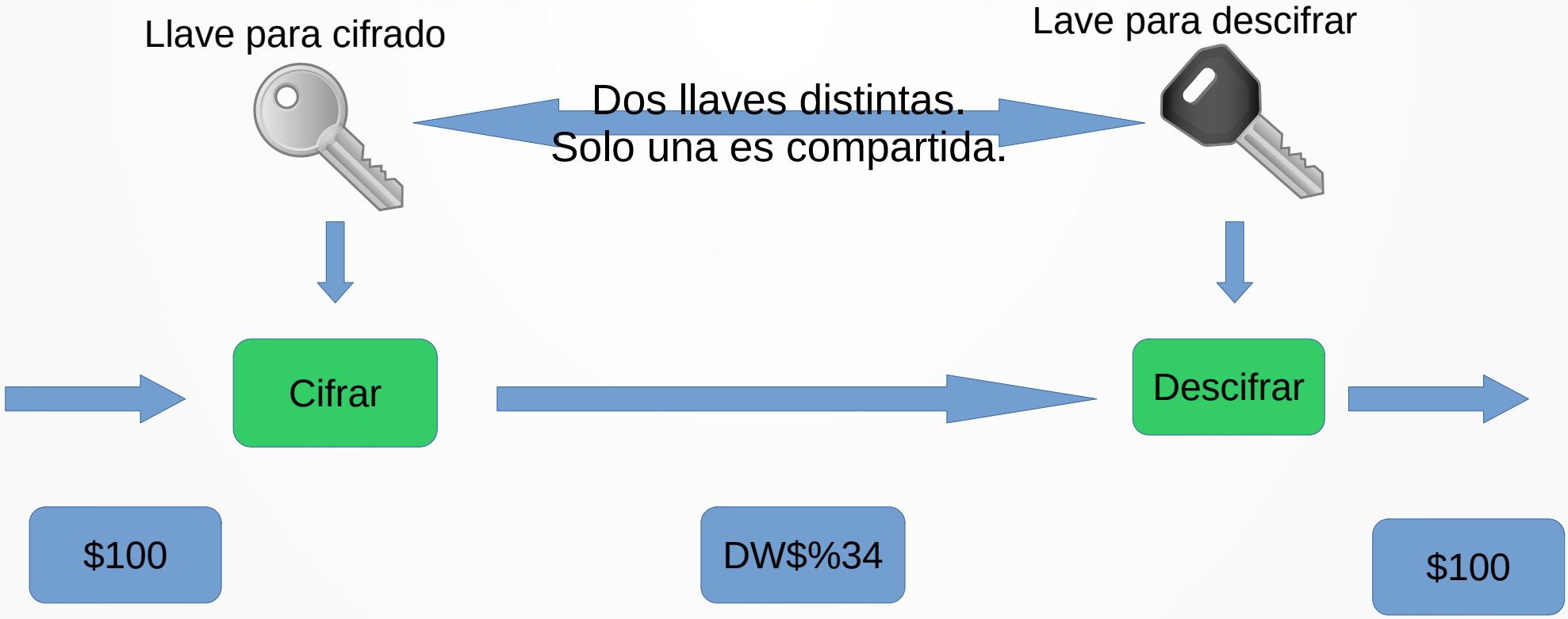
- Dos clases básicas de algoritmos de cifrado protegen las llaves
- Difieren en como utilizan la llave
  - Simétricos
    - Utilizan la “llave secreta” para cifrar y descifrar el mensaje
    - La llave debe ser compartida con anterioridad entre el origen y el destino
  - Asimétricos
    - Utilizan diferentes llaves para cifrar y descifrar el mensaje

Sistemas de cifrado (cont.)



Algoritmos de cifrado simétrico
Largo de llave típico: 80 a 256 bits
Ejemplos: DES, 3DES, AES, Blowfish

Sistemas de cifrado (cont.)



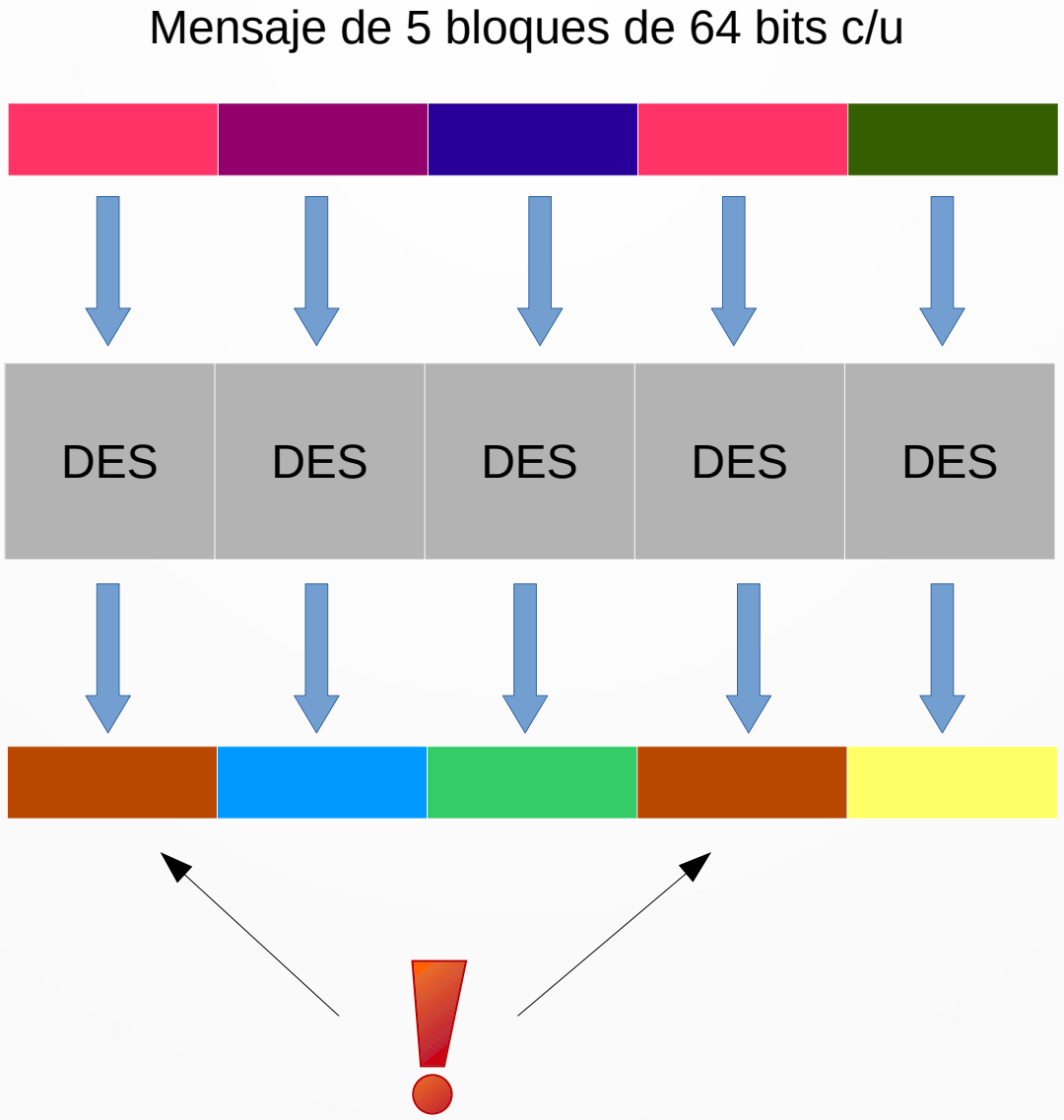
Algoritmos de cifrado Asimétrico
Largo de llave típico: 512 a 4096 bits
Ejemplos: RSA, ElGamal, DH

DES

- Data Encryption Standar
  - Algoritmo de cifrado simétrico
  - Generalmente en modo de bloques de 64 bits
    - ECB (Electronic Code Book)
    - CBC (Cipher Block Chaining)
  - Tamaño de llave de 56bits

DES (cont.)

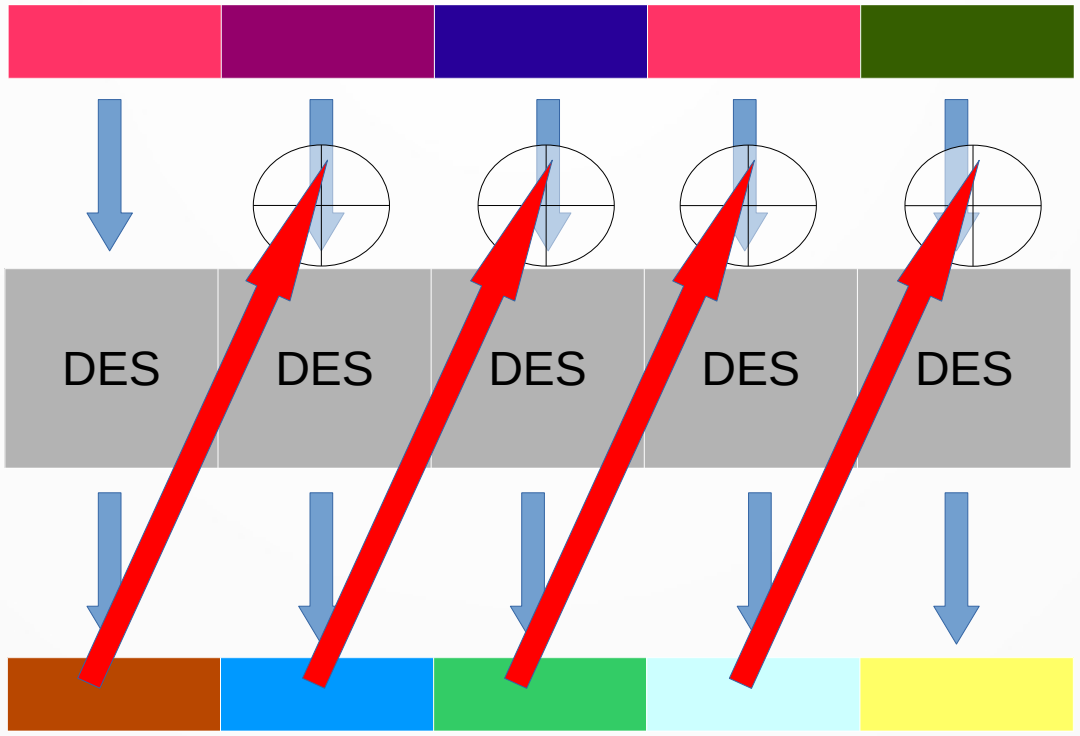
DES en modo ECB



DES (cont.)

DES en modo CBC

Mensaje de 5 bloques de 64 bits c/u



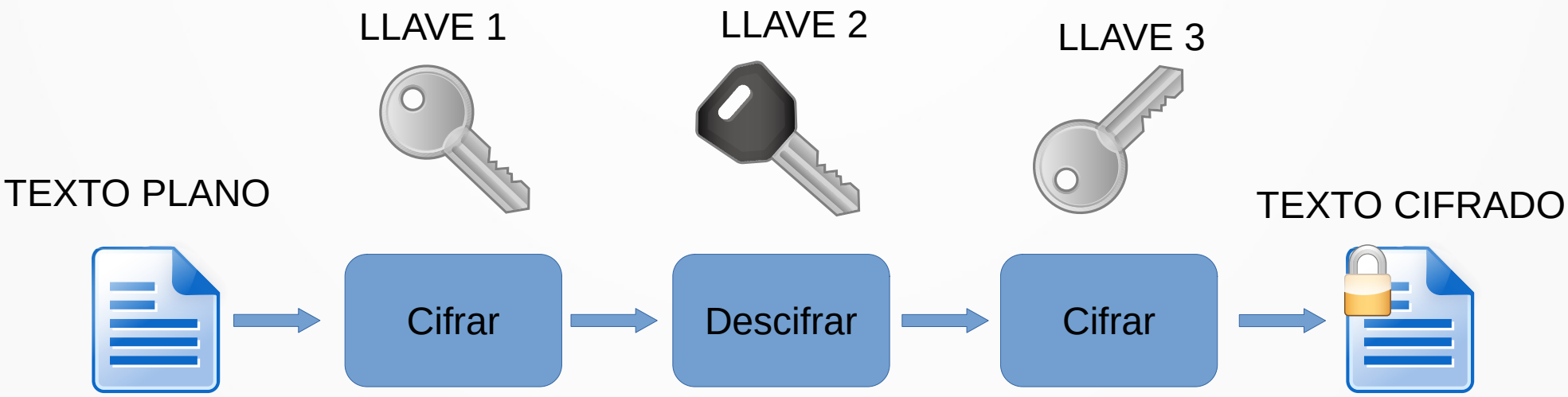
XOR bit a bit del bloque de texto con el resultado del bloque anterior

3DES

- La llave original de 56 bits resulta muy pobre para los ataques actuales
- En lugar de incrementar la llave (y modificar el algoritmo)
- Usar el mismo algoritmo 3 veces seguidas con diferentes llaves
- Llave de 112 a 168 bits

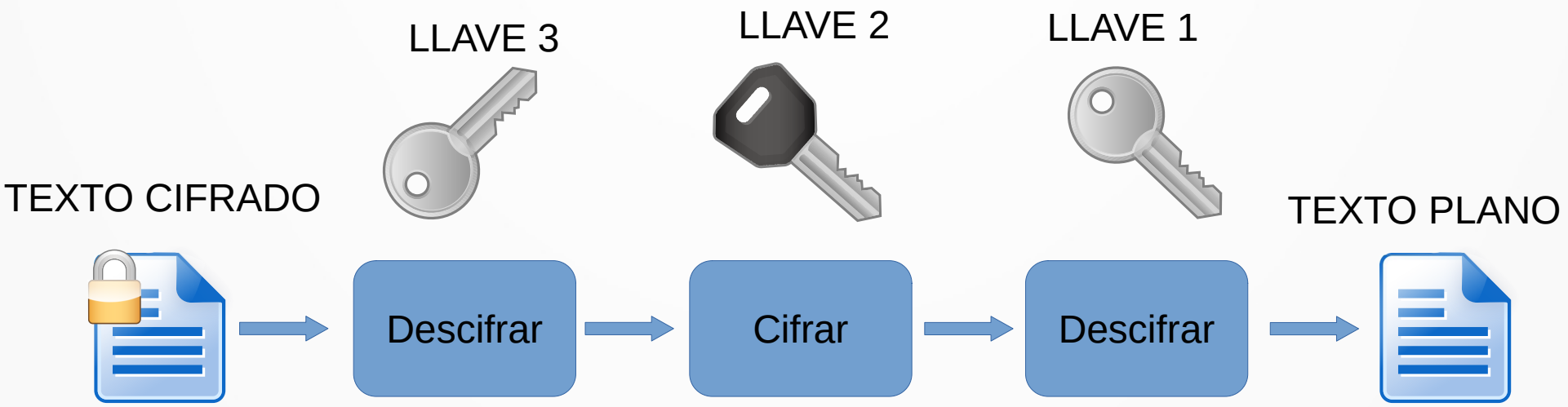
3DES (cont.)

CIFRAR



3DES (cont.)

DESCIFRAR



Ejercicio

- Realice el ejercicio ECB vs CBC
  - Descargue la imagen Tux.bmp
  - Utilice openssl para cifrar en modos ecb y cbc
  - Verifique los resultados
  - Puede ver la imagen cifrada?

AES

- Advanced Encryption Standard
  - NIST (National Institute of Standards and Technology) eligió el cifrado Rijndael en 2001 en el que se basa AES
  - Algoritmo de cifrado simétrico
  - Bloques de 128, 192 o 256 bits
  - Llaves de 128, 192 o 256 bits
  - https://formaestudio.com/rijndaelinspector/archivos/Rijndael_Animation_v4_eng-html5.html

DH

- Algoritmo Diffie-Hellman
  - Método de intercambio seguro de llaves
  - Se puede utilizar junto con algoritmos simétricos (llave compartida)
  - Permite que el origen y destino generen una llave compartida sin tener un intercambio previo de dicha llave

DH (cont.)

Alice		
Compartido	Secreto	Calculo
1 p=23		
g=5		
	2 a = 6	
		3 A = 5^6 mod 23 = 8

Bob		
Compartido	Secreto	Calculo
1 p=23		
g=5		
	2 a = 15	
		3 A = 5^15 mod 23 = 19

1- Alice y Bob se ponen de acuerdo en p (un numero primo) y en g (un numero base, generador)

2- Alice y Bob por su cuenta generan un numero secreto (a) y no lo comparten

3 – Alice y Bob por su cuenta calculan A = g^a mod p

DH (cont.)

Alice		
Compartido	Secreto	Calculo
p=23		
g=5		
	a = 6	
		A = 5^6 mod 23 = 8
4 A(bob) = 19	5	s = 19^6 mod 23 = 2

Bob		
Compartido	Secreto	Calculo
p=23		
g=5		
	a = 15	
		A = 5^15 mod 23 = 19
4 A(alice) = 8	5	s = 8^15 mod 23 = 2

4 – Alice y Bob intercambian el resultado de A

5- Alice y Bob por su cuenta calculan s = A(del otro)^a mod p

(s) es la llave secreta

Demo DH en python
http://neilrieck.net/dh_demo.html

LAB

- Laboratorio AES
  - Crear un archivo de texto
  - Crear una clave secreta
  - Cifrar el texto usando AES con la clave secreta
  - Generar un HMAC del texto cifrado
  - Enviar al destino el texto cifrado y su correpondiente hmac
  - Verificar autenticidad del mensaje cifrado recibido
  - Descifrar mensaje recibido
  - Comandos:
    - `openssl enc`